Here's your **Angular project setup without authentication guards** for the **Online Learning Platform**.

## **1. Create a New Angular Project (Non-Standalone)**

ng new online-learning-platform --no-standalone

cd online-learning-platform

## **2. Generate Modules and Components**

ng g module home --routing

ng g component home/home

ng g module courses --routing

ng g component courses/courses

ng g component courses/course-detail

ng g module profile --routing

ng g component profile/profile

## **3. Project Structure**

src/

├── app/

│ ├── home/

│ │ ├── home.module.ts

│ │ ├── home-routing.module.ts

│ │ ├── home.component.ts

│ │

│ ├── courses/

│ │ ├── courses.module.ts

│ │ ├── courses-routing.module.ts

│ │ ├── courses.component.ts

│ │ ├── course-detail.component.ts

│ │

│ ├── profile/

│ │ ├── profile.module.ts

│ │ ├── profile-routing.module.ts

│ │ ├── profile.component.ts

│ │

│ ├── app.module.ts

│ ├── app-routing.module.ts

│ ├── app.component.ts

## **4. Routing Configuration**

### app-routing.module.ts

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

const routes: Routes = [

{ path: '', redirectTo: '/home', pathMatch: 'full' },

{ path: 'home', loadChildren: () => import('./home/home.module').then(m => m.HomeModule) },

{ path: 'courses', loadChildren: () => import('./courses/courses.module').then(m => m.CoursesModule) },

{ path: 'profile', loadChildren: () => import('./profile/profile.module').then(m => m.ProfileModule) },

{ path: '\*\*', redirectTo: '/home' }

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

### courses-routing.module.ts

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { CoursesComponent } from './courses.component';

import { CourseDetailComponent } from './course-detail.component';

const routes: Routes = [

{ path: '', component: CoursesComponent },

{ path: ':id', component: CourseDetailComponent } // Dynamic Course ID

];

@NgModule({

imports: [RouterModule.forChild(routes)],

exports: [RouterModule]

})

export class CoursesRoutingModule { }

### profile-routing.module.ts (without authentication guard)

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { ProfileComponent } from './profile.component';

const routes: Routes = [

{ path: '', component: ProfileComponent }

];

@NgModule({

imports: [RouterModule.forChild(routes)],

exports: [RouterModule]

})

export class ProfileRoutingModule { }

## **5. Using** RouterLink **for Navigation**

### app.component.html

<nav>

<a routerLink="/home">Home</a> |

<a routerLink="/courses">Courses</a> |

<a routerLink="/profile">Profile</a>

</nav>

<router-outlet></router-outlet>

### **Courses List Example (**courses.component.html**)**

<ul>

<li \*ngFor="let course of courses">

<a [routerLink]="['/courses', course.id]">{{ course.name }}</a>

</li>

</ul>

### **This setup meets your requirements:**

✅ **Modules & Routing**  
✅ **Lazy Loading**  
✅ **RouterLink Navigation**  
✅ **Dynamic Course ID as Route Parameter**  
❌ **No Authentication Guards** (as requested)

Let me know if you need any modifications!